

APPLICATION FOR UNITED STATES LETTERS PATENT

FOR

**METHOD AND APPARATUS FOR PROCESSING LATENCY SENSITIVE ELECTRONIC  
DATA WITH INTERRUPT MODERATION**

Inventor(s): Linden Minnick  
Patrick L. Connor

Prepared by: Howard Skaist,  
Senior IP Attorney



Intel Corporation  
2111 N.E. 25th Avenue; JF3-147  
Hillsboro, OR 97124  
Phone: (503) 264-0967  
Facsimile: (503) 264-1729

"Express Mail" label number\_ ET 616076768 US \_

## METHOD AND APPARATUS FOR PROCESSING LATENCY SENSITIVE ELECTRONIC DATA WITH INTERRUPT MODERATION

5

### BACKGROUND

#### 1. Field

10 This disclosure relates generally to computer input/output (I/O) devices and interrupt moderation.

#### 2. Background Information

15 A computing platform or device is any hardware and/or software based system that includes the capability to perform logic and/or arithmetic operations, and may include the capability to transmit and receive data between itself and one or more additional computing platforms coupled via a transmission media. Computing platforms may include, without limitation, personal computers, laptop computers, servers, set-top boxes, hand-held devices, and numerous other types of systems. A computing platform may include code in the form of instructions, including but not limited to one or more device drivers or operating systems. 20 Computing platforms coupled to a transmission medium may include, for example, servers, personal computers, routers, switches, or hubs. Additionally, there are several different types of networks, including Wide Area Networks (WAN) and Local Area Networks (LAN) to cite two possible examples.

25 Computer networks are typically constructed by coupling two or more computing platforms to a data transmission medium, such as, for example, fiber optic cable, category 5 (CAT-5) network cabling, or wireless media, such as a wireless local area network (WLAN). These computing platforms typically access the transmission medium through an input/output (I/O) device, such as a network interface card (NIC) or local area network (LAN) on

5 motherboard (LOM), for example. I/O devices may comprise any device capable of routing, switching, repeating, receiving or passing data between one or more computing platforms or devices. An I/O device may include a network transmitter and/or receiver, and an I/O controller, for example. Electronic data is typically exchanged by using one or more types of data protocols, such as, for example, the transmission control protocol (TCP), which, as is well  
 10 known, is a protocol that operates within a multi-layered packet-based protocol, such as the Ethernet protocol. Details regarding TCP/IP may be obtained by referencing Request For Comments (RFC) 1180, January 1991. RFC 1180 may be obtained from the World Wide Web at the following URL: [www.freessoft.org](http://www.freessoft.org). The Ethernet protocol is defined in the Institute for Electrical and Electronics Engineers standard 802.3, 2000 edition, available from IEEE  
 15 standards, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ, 08855-1331. Additional information may be found on the World Wide Web at the following URL: <http://www.ieee.org>.

Recent developments in networking technology have resulted in increases in potential data transfer rates of electronic data across networks. In this context, electronic data refers to  
 20 one or more bits of digital data. These advances may result in network data transfer rates that exceed the data processing rates of computing platforms, which may ultimately result in a decrease in overall data transfer efficiency. A need exists for a method of increasing the data processing rate of computing platforms.

## 25 BRIEF DESCRIPTION OF THE DRAWINGS

Subject matter is particularly pointed out and distinctly claimed in the concluding portion of the specification. The claimed subject matter, however, both as to organization and method of operation, together with objects, features, and advantages thereof, may best be understood

by reference to the following detailed description when read with the accompanying drawings in which:

FIG. 1 is a schematic diagram of an I/O device and computing platform in accordance with one embodiment of the claimed subject matter.

FIG. 2 is a flowchart illustrating one embodiment of the claimed subject matter.

FIG. 3 is a diagram of a computing platform in accordance with one embodiment of the claimed subject matter.

# DETAILED DESCRIPTION

In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the claimed subject matter. However, it will be understood by those skilled in the art that the claimed subject matter may be practiced without these specific details. In other instances, well-known methods, procedures, and components have not been described in detail so as not to obscure the claimed subject matter.

Typically, although the claimed subject matter is not limited in this respect, computing platforms coupled to a transmission medium are coupled through an input/output (I/O) device, such as a network interface card (NIC), which may alternatively be referred to as a server adapter, network adapter, or media access card, but the claimed subject matter is not limited in this respect. There are many types and categories of NICs, and the claimed subject matter is not limited to any particular type of NIC or I/O device, and may include external NICs, onboard NICs, or peripheral NICs, without limitation. One such NIC may comprise an Ethernet Media Access Controller (MAC), such as an Intel® 82543 GC Gigabit Controller, available from Intel Corporation, Santa Clara, California. Such NICs typically manage data transfer between a computer and a network, and typically operate using a particular type of protocol. There are

5 many versions of protocols that may be used to practice the claimed subject matter, including Ethernet and Gigabit Ethernet, which, as is well known, relate to a 10 and 1000 Megabits per second (Mb/s) rate of electronic data transfer, respectively, although it will be understood that there are many other types of media access control protocols that may be used in accordance with the claimed subject matter. The Gigabit Ethernet protocol is defined for data transfer over  
10 fiber optic cable in the Institute of Electrical and Electronics Engineers (IEEE) standard 802.3z-1998, and for data transfer over CAT-5 cable in IEEE standard 802.3ab-1999. Details regarding this protocol can be obtained from IEEE Standards, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331. Additional information can be found on the World Wide Web at the following URL: <http://www.gigabit-ethernet.org>. NICs such as the type previously described  
15 typically operate by utilizing at least one device driver. In one embodiment, a device driver comprises a software program that may reside on a computing platform, for example, and at least partially provides commands to the NIC and manages the operation of the NIC and how the NIC interacts with other components or systems coupled to a network.

20 FIG. 1 illustrates one embodiment of an I/O device coupled to a computing platform, although the claimed subject matter is not limited in this respect. 100 is an example of a system comprising an I/O device 104, coupled to a network 102. I/O device 104 may, in one embodiment, be coupled to a network 102 by or through transmission media 106. As stated previously, transmission media may include, for example, fiber optic cable, category 5 (CAT-5)  
25 network cabling, or wireless media, such as a wireless local area network (WLAN). Transmission media 106 may, in one embodiment, be coupled to I/O device 104 through a physical interface 108. In one embodiment, I/O device 104 is coupled to a computing platform 120. I/O device 104 may, in one embodiment, be coupled to computing platform 120 by a platform bus 118. There are many different types of platform buses that may be used in  
30 accordance with the claimed subject matter. One such platform bus comprises a bus that

5 complies with a peripheral component interconnect (PCI) version 2.2 specification or a PCI-X version 1.0a specification compliant bus. Information regarding the PCI version 2.2 specification and/or PCI-X version 1.0a specification may be obtained from the PCI special interest group (PCI-SIG), 5440 SW Westgate Dr., 217, Portland, Oregon 97221. Additional information may be obtained at the PCI-SIG website on the World Wide Web at the following URL: [www.pcisig.com](http://www.pcisig.com).

10 Platform bus 118 may, in one embodiment, be coupled to I/O device 104 through a bus interface 116. It will, of course be understood that this is just one embodiment, and I/O device 104 may be coupled to computing platform 120 via a transmission medium other than a platform bus and still be within the scope of the claimed subject matter.

15 As is well-known, network architectures are typically designed by partitioning complex tasks into one or more subtasks, also known as layers. A layer typically communicates with one or more peer layers in a format referred to as a protocol. In this context, a set of related protocols may be referred to as a protocol stack. A protocol, in this context, comprises a set of rules and conventions that may be employed by one layer to communicate with another layer in an associated network. An example of a protocol having layers may include the transmission control protocol (TCP). TCP operates, at least in part, to send and/or receive fragments of electronic data across a network. A fragment of electronic data, in this context, comprises a temporally contiguous sequence of digital data pulses. A packet, in this context, comprises one or more fragments of electronic data. In the process of sending and/or receiving data packets, 20 TCP will typically utilize an acknowledgement, or ACK, to confirm that a particular fragment or packet of electronic data was successfully delivered to a particular node on an associated network. As is well-known, once an I/O device operating within a TCP environment transmits a data packet, typically the I/O device that receives the packet will send an ACK to confirm delivery. If an ACK is not received by the sending I/O device within a particular period of time, 25 also referred to as a timeout, the sending I/O device will typically re-send at least a portion of

5 the electronic data. The elapsed time between a packet being transmitted by an I/O device and an ACK being received by the sending I/O device is commonly referred to as round trip time (RTT). Typically, in a TCP environment, electronic data transmission rates are affected by the RTT. As RTT increases, throughput or the electronic data transfer rate typically decreases. Electronic data transmission rates in a TCP environment are also typically affected by the TCP  
10 window size. Window size, as is well known, is the amount of data that may typically be sent by an I/O device before it waits to receive an ACK, and may affect data flow or data congestion across a network. Typically, if an ACK has not been received by a transmitting I/O device subsequent to the window size being met, the transmitting I/O device will not send additional data. ACK packets are one type of latency sensitive packets or data fragments, meaning, in this  
15 context, that these packets have the capability to affect the data transfer rate in the manner described previously, if not processed in an efficient or timely manner, although the claimed subject matter is not limited to just ACK packets. Again, this is just one type of latency sensitive packets or data fragments.

20 Latency sensitive fragments of electronic data, in this context, comprise one or more fragments of electronic data that may be more time sensitive than typical fragments of electronic data. This may include, for example, ACK packets, as stated previously. This may additionally include fragments of electronic data designated for streaming media applications, or may include one or more data packets that have a priority designation. In this context, time sensitive  
25 refers to a fragment of electronic data that may have an affect on the performance of a computing platform or software application if not received and processed within a particular period of time, or within a timely fashion, although the claimed subject matter is not limited in this respect. Examples of a designation of priority in a packet may include, for example, a designation in the priority field, or the precedence or type of service (ToS) fields in a packet that  
30 complies with the Ethernet protocol. Priority status may not be designated in a particular field,

5 but the type of packet may be detected and identified as high priority by the receiving device based, at least in part, on the packet type. Priority designations are well known in the art, and the claimed subject matter is not limited to just these examples of priority designations or time sensitive fragments of electronic data.

10 As is well known, computing platforms typically utilize interrupts to shift control or partially shift control of a processor, and may shift which task is being processed by a processor, or alter the particular manner in which a task is processed. For example, an interrupt may be asserted in order to stop or slow down the processing of a particular task, and allow it to at least partially process another task, such as, for example, processing data that has been received by an I/O device. As stated previously, when an interrupt is asserted, the task being processed by the processor prior to the interrupt will typically cease. Typically, data processing rates may be affected by the number and frequency of interrupts used on a processor, as typically, asserting interrupts will stop the processing of certain functions, and start the processing of other functions. Computing platforms may have one or more interrupt schemes.

15 An interrupt scheme, in this context, refers to a method, plan or criteria for when and how to assert or allow the assertion of interrupts. Interrupt moderation, in this context, refers to the assertion or deferral of interrupts based at least in part on a particular interrupt scheme. For example, an interrupt scheme may be designed to assert or allow the assertion of one or more interrupts of a computing system processor if a particular device requests that data be

20 processed, such as, for example, an I/O device requesting that a computing system processor process data that the I/O device has received. Alternatively, an interrupt scheme may be designed to allow a device to assert one or more interrupts after a certain number of similar requests have been received, after a passage of time, or any number of other factors, although the claimed subject matter is not limited in this respect. An interrupt scheme may be designed  
25 based on one or more factors that may include, for example, asserting one or more interrupts to  
30



5 improve processor usage, asserting one or more interrupts to process data received by an I/O device as quickly as practical, or to assert one or more interrupts based on importance of the time sensitivity of the data which is requested to be processed, although the claimed subject matter is not limited in this respect.

10 In one embodiment, the interrupt scheme of an associated computing platform may be designed to assert or allow the assertion of one or more interrupts based on the contents of a fragment of electronic data. In this embodiment, this may include the receipt of a fragment of electronic data that is latency sensitive, as stated previously. In one embodiment of the claimed subject matter, a method and/or apparatus for processing latency sensitive data by using  
15 interrupt moderation may comprise an I/O device as shown in FIG 1. I/O device 104 may be comprised of several components, including but not limited to a physical interface 108, a bus interface 116, a parsing module 112, a memory 114, and an interrupt moderation management module 110. A physical interface, in this context, refers to an interface permitting a transmission media, such, as for example, fiber optic cable, to be coupled to the I/O device, and may also  
20 include the receiver of the electronic data within the I/O device. An interrupt management module, in this context, refers to a device or software capable of at least partially generating interrupts, and/or deferring interrupts, based at least in part on an interrupt scheme, which may itself be based on particular criteria such as packet type, for example, as explained in more detail later.

25 In operation, a fragment of electronic data may be received by I/O device 104, from a node on an associated network 102. As stated previously, this fragment of electronic data may be at least partially received at physical interface 108. The physical interface may, in this embodiment, pass at least a portion of the electronic data to the parsing module 112, although  
30 in alternative embodiments, the data fragment may be passed to other components of the I/O

5 device such as, for example, memory 114. The parsing module, in this embodiment, performs functions, such as, substantially determining the characteristics of a fragment of electronic data, such as, for example, packet type, if the fragment of electronic data comprises a data packet. The parsing module may alternatively, or additionally, determine the size of the fragment, if, for example, the fragment of electronic data comprises an ACK, or if, for example, the fragment of electronic data has one or more priority designations or otherwise comprises a latency sensitive  
10 fragment. Parsing modules are well known in the art, and the claimed subject matter is not limited to any particular type of parsing module or to any device or software that utilizes one or more parsing functions as a separate module. Parsing module 112 may, in this embodiment, pass information about the characteristics of a fragment of electronic data to interrupt  
15 management module 110. Subsequently, the parsing module may, in this embodiment, pass at least a portion of the fragment of electronic data to memory 114. Interrupt management module 110 may, in this embodiment, moderate the interrupt scheme of the I/O device based at least in part on the characteristics of a fragment of electronic data. For example, if parsing module 112 determines that a fragment of electronic data comprises an ACK packet, the parsing module  
20 may pass this information to interrupt management module 110. Interrupt management module 110 may, in one embodiment, moderate an interrupt so that the I/O device may generate an interrupt of processor 124 of an associated computing platform 120 substantially immediately after receipt of the ACK, so computing platform 120 may then process at least a portion of the fragment of electronic data.

25

FIG 2 illustrates one embodiment of a method and/or apparatus for processing latency sensitive data by using interrupt moderation in accordance with the claimed subject matter. In this particular embodiment, 202 includes an I/O controller receiving a fragment of electronic data, which, in this embodiment, comprises a packet. 204 may be performed by a parsing  
30 module, in this embodiment, and the parsing module may inspect the packet to determine if it is

5 an ACK packet. If the packet is not an ACK packet, 206 includes providing an instruction to the I/O device to moderate an interrupt by deferring an interrupt of an associated computing platform processor. If, however, the parsing module determines in 204 that the fragment of electronic data comprises an ACK packet, 208 includes providing an instruction to the I/O device to then assert an immediate or substantially immediate interrupt of an associated  
10 computing system processor, although, of course, the claimed subject matter is not limited in this respect. For example, if at 204 it is determined that the fragment of electronic data comprises an ACK packet, the I/O device may defer interrupting an associated computing system processor, or defer it for a particular period of time, or until more ACK packets are received. These are just a few examples of alternative embodiments. Additionally, as stated  
15 previously, the I/O device may base interrupt moderation on other criteria of fragments of electronic data, such as packet size, packet priority, number of packets received, or amount of data received over a particular period of time, but the claimed subject matter is not limited to these examples. Additionally, 208 may include an instruction to the I/O device to then assert an immediate or substantially immediate interrupt after the expiration of a particular period of time, but, again, the claimed subject matter is not limited in this respect.  
20

It will, of course, be understood that this is just one embodiment of the claimed subject matter, and alternative methods of interrupt moderation may be implemented in accordance with the claimed subject matter. For example, in an alternative embodiment, I/O device 104 of FIG 1  
25 may receive a fragment of electronic data from an associated network 102. As stated previously, this fragment of electronic data may be at least partially received at the physical interface 108. The physical interface may, in this alternative embodiment, pass at least a portion of the electronic data to memory 114. Interrupt management module 110 may, in this embodiment, moderate the interrupt functions of the I/O device depending at least in part on the  
30 characteristics of a fragment of electronic data, such as the size. For example, if interrupt

5 management module 110 determines that the fragment of electronic data is below a certain size, such as 80 bytes, for example, interrupt management module 110 may then moderate the interrupt and generate an interrupt of processor 124 of an associated computing platform 120 after one or more of these data fragments have been received. Additionally, the interrupt management module may generate an interrupt of processor 124 after a certain amount of  
10 memory module 114 has been utilized to store fragments of electronic data. For example, interrupt management module 110 may generate an interrupt of processor 124 after 10 Kilobytes of fragments of electronic data have been stored in memory module 114, although the claimed subject matter is not limited in this respect. In another alternative embodiment, the interrupt management module may moderate one or more interrupts based at least in part on  
15 characteristics of a fragment of electronic data, such as a priority designation that may be designated by at least a portion of the fragment of electronic data, as explained previously. For example, if a fragment of electronic data is designated as high priority, this information may be identified at least in part by parsing module 112, and this information may be passed at least partially to interrupt management module 110. Interrupt management module 110 may, in this  
20 alternative embodiment, generate an interrupt of processor 124 at a time subsequent to the receipt of the fragment of electronic data, and instruct the computing platform 120 to process at least a portion of the fragment of electronic data. Additionally, an interrupt management module may track the number of interrupts per unit of time that have been generated, and if a predetermined maximum number of interrupts per unit of time has been reached, an interrupt  
25 may be deferred for a particular period of time, even if a fragment of electronic data comprises a high priority packet.

In yet another alternative embodiment, interrupt moderation or an interrupt scheme may be controllable at least in part by a user. In this yet another alternative embodiment, a user may  
30 establish criteria for determining when to interrupt an associated computing platform processor

5 124 of FIG 1 to process at least a portion of electronic data, such as, for example, elapsed time after receipt of a fragment of electronic data, particular characteristics of fragments of electronic data, or quantity of fragments of electronic data received, or any combination thereof, but the claimed subject matter is not limited to just these criteria. A user, in this alternative embodiment, may program the apparatus for processing latency sensitive electronic data by using interrupt  
 10 moderation. This programming may take place through a user interface, for example. The interface, in one embodiment, may comprise a user interface or graphical user interface for an associated computing platform, as shown in FIG 3. The user, in this alternative embodiment, may enter criteria, such as the criteria stated previously, and substantially program an associated I/O device 104 to interrupt an associated computing platform processor 124 based  
 15 on one or more of the above-mentioned criteria. The user may, in this yet another alternative embodiment, program the I/O device 104 to moderate one or more interrupts of an associated computing platform processor 124 prior to the receipt of a fragment of electronic data, after the receipt of a fragment of electronic data, or any other time, but the claimed subject matter is not limited in this respect.

Embodiments of the claimed subject matter may be implemented in hardware, firmware or software, or any combination thereof. Embodiments of the claimed subject matter may be implemented as a computer program executing on a computing system, comprised of at least  
 25 one processor, a data storage system, which includes volatile and non-volatile memory and/or storage elements, at least one input device, and at least one output device. FIG. 3 is a block diagram of a system 300 of one embodiment of the claimed subject matter. System 300 includes a processor 302 that processes data signals, and may comprise, for example, a PENTIUM®III or PENTIUM® 4 microprocessor available from Intel Corporation. Processor 302  
 30 may be coupled to a processor bus 304 that transmits data signals between processor 302 and

5 other components in system 300. System 300 may include a memory 306. Memory 306 may store instructions and/or data represented by data signals that may be executed by processor 302. The instructions and/or data may comprise code for performing any and/or all of the techniques, for example. Memory 306 may also store additional software previously described and/or data in accordance with this embodiment, including device drivers for associated  
10 components. The software alternatively may be stored on a data storage device 312, such as, for example, a floppy disk drive, a read only memory (ROM), a CD-ROM device, a flash memory device, a digital versatile disk (DVD), or other storage device. A cache memory 310 may reside within processor 302 that stores data signals received from memory 306. A network controller 308 may be coupled to processor bus 304. A user interface 314 may be coupled to the  
15 processor bus 304, or execute on a component coupled to bus 304, and may typically allow a user to gain access to one or more components of system 300. For purposes of this application, a processing system embodying dynamic coalescing components includes any system that has a processor, such as, for example, a digital signal processor (DSP), a microcontroller, an application specific integrated circuit (ASIC), or a microprocessor.  
20

While certain features of the claimed subject matter have been illustrated as described herein, many modifications, substitutions, changes, and equivalents will now occur to those skilled in the art. It is, therefore, to be understood that the appended claims are intended to  
25 cover all such embodiments and changes as fall within the true spirit of the claimed subject matter.